PNAS RESEARCH ARTICLE | BIOPHYSICS AND COMPUTATIONAL BIOLOGY

Taxonomic classification of DNA sequences beyond sequence similarity using deep neural networks

Florian Mock[a,1], Fleming Kretschmer[b,1], Anton Kriese[c], Sebastian Böcker[b], and Manja Marz[a,d,e,f,2]

Edited by Molly Przeworski, Columbia University, New York, NY; received December 15, 2021; accepted July 2, 2022

Taxonomic classification, that is, the assignment to biological clades with shared ancestry, is a common task in genetics, mainly based on a genome similarity search of large genome databases. The classification quality depends heavily on the database, since representative relatives must be present. Many genomic sequences cannot be classified at all or only with a high misclassification rate. Here we present BERTax, a deep neural network program based on natural language processing to precisely classify the superkingdom and phylum of DNA sequences taxonomically without the need for a known representative relative from a database. We show BERTax to be at least on par with the state-of-the-art approaches when taxonomically similar species are part of the training data. For novel organisms, however, BERTax clearly outperforms any existing approach. Finally, we show that BERTax can also be combined with database approaches to further increase the prediction quality in almost all cases. Since BERTax is not based on similar entries in databases, it allows precise taxonomic classification of a broader range of genomic sequences, thus increasing the overall information gain.

taxonomic classification | deep learning | meta genome

Significance

The correct assignment of DNA sequences to their origin is an important task. However, only a fraction of all species are available in today's databases and thus easily assignable. Therefore, we present a method that is particularly good at classifying sequences for which there are no closely related species in databases. For this purpose, we use a deep learning approach to learn, at first, the "language" of DNA to subsequently distinguish the "language" structure of different groups of organisms, for example, bacteria and viruses. Using this approach, we achieve comparable quality to previous methods for sequences with close relatives in the database and superior quality for new species.

How do we know what kind of organisms we have sequenced?

This question seems, at first, rather strange, as, traditionally, DNA sequencing was mostly performed on cultivated cells or viral strains/isolates. However, in recent years, this has become a common problem, especially due to metagenomics, where genetic material is directly recovered from environmental samples and includes unknown compositions of organisms. To answer this question, we need to classify the taxonomic origin of the sequences.

For this task of taxonomic classification, it became common to use a homology-based approach of DNA/RNA sequences queried against databases. Such approaches achieve a high level of precision and can even determine the exact species when the genome is already known and present in the database.

However, many query sequences cannot be classified at all, and,, therefore only a low recall is achieved. One of the reasons is that only a fraction of all species (at most, 14% of all eukaryotic terrestrial and 9% of all ocean species) have been described (1). Furthermore, this estimate ignores bacteria and viruses for which the situation is likely way worse (2). The number of reference genomes in current comprehensive databases, such as RefSeq (117,030 as of March 11, 2022), likely represents less than 5.319% of all species (3), which is a rather optimistic estimation (4, 5). As a result, taxonomically related organisms and, therefore, likely genomically similar sequences are missing, and the classification with homology-based methods fails. The extent of this problem highly depends on the origin of the sample, the taxonomic level to classify, and the database. For example, the number of unclassifiable sequences for the phylum rank varies between 25% and 90%, using biogas reactor samples (6).

On the technical level, most taxonomic classification tools use either local alignments, k-mers, Burrow–Wheeler transformations, minimizers, or hybrid methods (7–17).

Using local alignments is a very precise but relatively slow method. These methods typically need a seed region with high similarity, typically resulting in a limited recall, meaning that only a portion of the samples can be predicted. The usage of k-mers or minimizer requires even wider regions with high similarity, which can result in an even lower recall and precision. However, k-mers and minimizer are significantly faster (18). Overall, the taxonomic classification quality depends mainly on the quality of the database and less on the methodology of the database usage (18).

Recently, multiple deep neural network (DNN) approaches were developed to overcome some of the described limitations. Instead of having to rely on similar sequences being present in a database, deep learning methods allow modeling complex dependencies between the data and the target variable, in our case, the DNA sequences and corresponding taxonomic class. Typically, for deep learning methods, however, interpretability

Author affiliations: [a]RNA Bioinformatics and High-Throughput Analysis, Friedrich Schiller University Jena, 07743 Jena, Germany; [b]Bioinformatics, Friedrich Schiller University Jena, 07743 Jena, Germany; [c]Institute for Computer Science, Freie Universität Berlin, 14195 Berlin, Germany; [d]Bioinformatics Core Facility, Friedrich Schiller University Jena, 07743 Jena, Germany; [e]Institute for Computer Science, Leibniz Institute for Age Research – Fritz Lippman Institute, 07745 Jena, Germany; and [f]European Virus Bioinformatics Center, 07743 Jena, Germany

Author contributions: F.M., F.K., and M.M. designed research; F.M., F.K., and A.K. performed research; S.B. and M.M. contributed new reagents/analytic tools; F.M. and F.K. analyzed data; and F.M., F.K., S.B., and M.M. wrote the paper.

The authors declare no competing interest.

This article is a PNAS Direct Submission.

[1]F.M. and F.K. contributed equally to this work.

[2]To whom correspondence may be addressed. Email: manja@uni-jena.de.

This article contains supporting information online at https://www.pnas.org/lookup/suppl/doi:10.1073/pnas.2122636119/-/DCSupplemental.

Published August 26, 2022.

is often not an easy task. One example of a deep learning approach is DeepMicrobes (19), which is specifically designed for taxonomic classification of microbial data from human gut; however, it can be used as a generic classifier for any DNA classification task. It is important to mention that DeepMicrobes relies on k-mer embeddings ($k = 12$), similar to nonmachine learning methods which handle sequence data.

Here, we present the tool BERTax for classification of DNA sequences on three different taxonomic levels, superkingdom (archaea, bacteria, eukaryota, and viruses), phylum, and genus. The fundamental novelty is to assume DNA is a "language" and to classify the taxonomic origin based on this language understanding rather than by local similarity to known genomes in a database.

As a result, we obtain a classifier not subjected to the typical restrictions of comparable tools: BERTax is not limited to coding regions, or specific superkingdoms like bacteria or viruses, but can potentially classify any genome region, from any of the four superkingdoms, and does not require similar sequences in the database.

At the core, BERTax is based on the state-of-the-art natural language processing (NLP) architecture BERT (bidirectional encoder representations from transformers) (20), which is adapted for the task of taxonomic classification via additional layers (21). Being standard for BERT, we 1) pretrained the DNN and 2) fine-tuned the DNN. In this approach, to classify a query sequence, less exact representatives are needed in the training data, since the training sequences are not memorized per se (20).

We developed three different DNN architectures: the 1) flat, 2) nested, and 3) all-in-one architectures, which we compared with each other. The latter and best-performing architecture has been compared with common database approaches and the deep learning approach DeepMicrobes. The performance of BERTax can further be increased by combining a precise database approach, like MMseqs2, with BERTax.

In raw numbers (macro average precision [AveP]), the performance improvement gained with BERTax is most visible for sequences from unknown genera, where an increase from 76.91 to 87.57% on the level of superkingdoms and from 50.51 to 60.34% on the phylum level can be achieved, compared with DeepMicrobes, both in combination with MMseqs2. When comparing standalone approaches, BERTax increases the macro AveP from 67.47 to 90.06% on the superkingdom level and from 43.44 to 54.10% on the phylum level, compared with MMseqs2 taxonomy*. But also, when closely related species are used for training, BERTax classifies genomic sequences, on average, with a slightly higher precision than previous tools.

Results

BERTax is based on the DNN encoder architecture BERT, that relies on a transformer employing the mechanism of self-attention (20). Self-attention is a method determining, autonomously, which parts of the inputs are relevant to each other. This enables the transformer architecture to process the sequential data not in a predefined order, enabling a faster training process and therefore the recognition of even more complex interrelationships in the same time (22).

The training process of BERTax is split into two parts: First, a BERT model is pretrained in an unsupervised manner (Fig. 1*A*), meaning that the target variables—which sequence belongs to which taxonomic classes—are not known to the model, with the goal of learning the general structure of the genomic DNA "language." The pretrained BERT model is then combined with taxonomy layers and fine-tuned on the specific task of predicting taxonomic classes (Fig. 1*B*).

The unsupervised pretraining was performed on a dataset based on 1 million genomic fragments of 1,500 nt for each of the four superkingdoms—archaea, bacteria, eukaryotes, and viruses—obtained from National Center for Biotechnology Information (NCBI) refseq genomes (23). These fragments were filtered by sequence identity of 80%, resulting in ∼2.5 million (2,492,474) samples in total (see *Data*).

For fine-tuning, three distinct evaluation scenarios are considered, represented by a different composition of the training and testing datasets: For the distantly related dataset, samples in

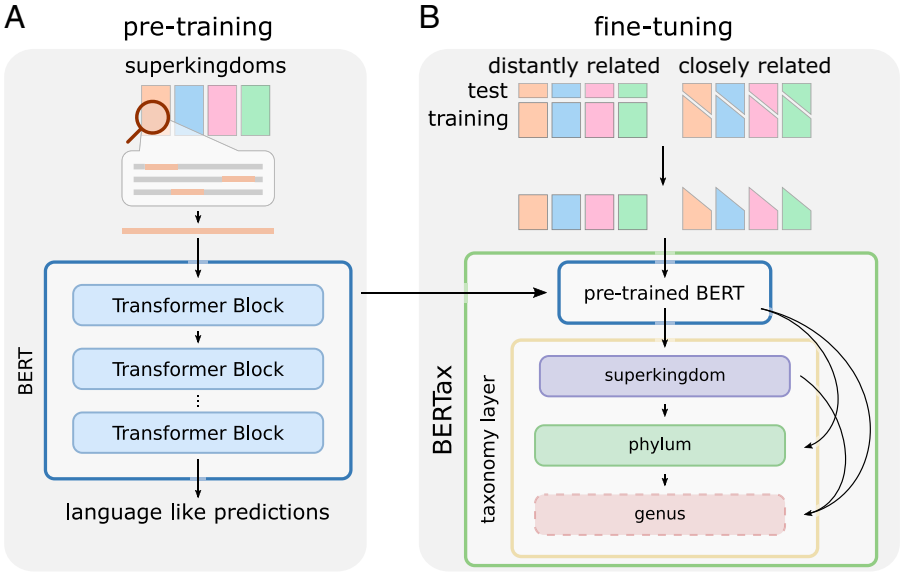

Fig. 1. (*A*) The pretraining step uses random genome fragments of all four superkingdoms to pretrain a BERT model on DNA analogous to training on a natural language. (*B*) The fine-tuning step uses, as input, either a training set with genome fragments of different genera compared to the test set (distantly related) or a dataset without this condition (closely related). These inputs are processed by BERTax. BERTax uses the pretrained BERT model and three taxonomy layers. The taxonomy layers use our all-in-one architecture which uses the output of the pretrained BERT model and all higher taxonomic ranks to predict superkingdom, phylum, and—only in the final model—also the genus of an input sequence.

the test and training sets are not allowed to be from the same taxonomic genus; this restriction is lifted for the closely related dataset. The final dataset corresponds to an extension of the closely related dataset, simply containing a higher number of samples to be used in the final BERTax model.

We tested three different architectures based on the pretrained BERT model, which differ in the adaptations of the output layers (*SI Appendix*, Fig. S1). The flat architecture (*SI Appendix*, Fig. S1*A*) directly predicts the lowest taxonomic rank. The nested architecture (*SI Appendix*, Fig. S1*B*) consists of multiple independent BERT models arranged in a tree-like manner: The root model predicts the highest taxonomic rank, and then the sample is passed to the model of the subclass with the highest probability. The all-in-one architecture (*SI Appendix*, Fig. S1*C*) is a single BERT model that simultaneously predicts all taxonomic ranks by providing the prediction of higher taxonomic ranks to predict lower taxonomic ranks.

Best Architecture Based on Simultaneous Classifications of Taxonomic Ranks. We used two different evaluation datasets (closely and distantly related dataset) for which we determine the accuracy of each architecture for predicting superkingdom and phylum (*SI Appendix*, Table S1). To prevent biasing the metric due to unbalanced data, we calculate the accuracy for each superkingdom class and use the mean over all superkingdom classes.

Interestingly, the all-in-one architecture has the highest accuracy independent of the dataset: for the closely related dataset, 94.78% for superkingdom and 85.55% for phylum; 88.95% and 60.10%, respectively, for the distantly related dataset. This is, on average, 2.04% more than the flat architecture and 7.22% more than the nested architecture (*SI Appendix*, Table S1).

The all-in-one architecture provides the predicted likelihood of all classes of higher taxonomic ranks to lower prediction layers. This is advantageous compared to the flat architecture, as higher taxonomic classes indicate which subset of lower taxonomic classes is likely to contain the correct prediction. However, unlike the nested architecture, the higher taxonomic class classification does not categorically exclude some lower taxonomic classes. This has several advantages. Misclassification of higher taxonomic ranks (e.g., superkingdom) does not inevitably prevent the correct prediction of lower ranks. Further, the training of lower taxonomic ranks can also adjust the weights and biases of the neural network.

In this way, it is possible to identify not only features that generalize well to an entire class, such as bacteria or eukaryotes, but also features of subgroups (*SI Appendix*, Fig. S2). These subgroups are expected to be clades of lower taxonomic rank. As a result of better subgroup prediction, the prediction of higher taxonomic ranks advances as well. For further analysis, we only use the all-in-one architecture.

Comparison to Other Classification Approaches. The database taxonomic classification approaches compared against either directly predict the taxonomic origin of a query (Kraken2, sourmash) or find the most similar sequence in a reference database (MMseqs2 and minimap2), which can also be considered a prediction of taxonomic origin. A version of MMseqs2 which additionally provides classification confidence, MMseqs2 taxonomy*, is also compared against. It has to be noted that all methods compared against have exactly the same reference or training data available as BERTax, to allow for a meaningful comparison. For approaches that normally include whole genomes in their reference data, this presents a necessary deviation from the designed use case for these methods. In particular, database approaches can be seen as reliant on homologous sequences in the reference data, in order to search for matches most similar to the query sequence. Moreover, exact matches between query and reference data (i.e., samples occurring in both training and testing data) are generally possible. While homologous segments between sequences in training and testing data independent of species taxonomy are considered valid for the evaluation of BERTax (see *SI Appendix*, Table S5 for an evaluation of how the presence of sequences with shared homology influences performance), duplicate samples in training and testing data are not permitted in this evaluation. Samples of the testing data with an (almost) identical counterpart in the training data would allow BERTax to avoid the problem of identifying taxonomy-specific features and simply remembering the duplicate samples, which could be considered a case of data leakage. Thus, having whole reference genomes present in the training data is not possible. The AveP for each tool is visualized in Fig. 2, and exact values are listed in Table 1.

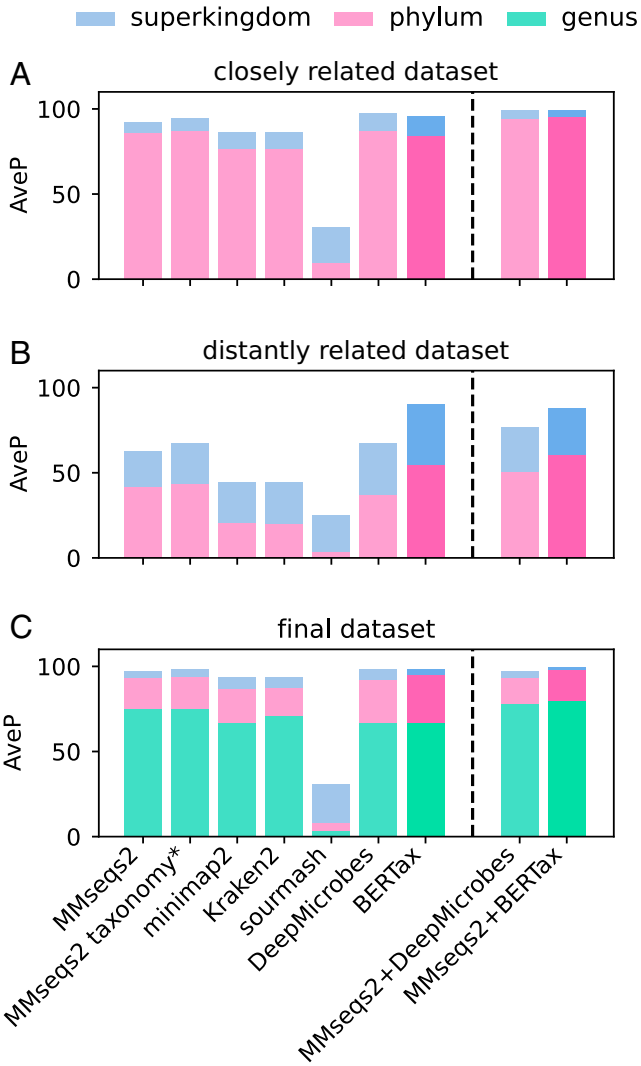

Fig. 2. Visualization of the macro AveP values on all three datasets. (*A*) In the closely related dataset, samples in the test set and the reference database (training set) can be from the same genus. (*B*) In the distantly related dataset, samples in the test set do not have closely related (identical genus) samples in the reference database. (*C*) The final dataset can contain samples in the test set and reference database of the same genus, like the closely related dataset, but comprises more samples. We queried by either superkingdom, phylum, or genus (only final dataset). MMseqs2 taxonomy*: We used MMseqs2 with the parameters of MMseqs2 taxonomy (–e-profile 0.001, -e 1).

Table 1. Comparison of the macro AveP on all three datasets

	Dataset						
	Closely related		Distantly related		Final		
Tool	Supk.	Phyl.	Supk.	Phyl.	Supk.	Phyl.	Genus
MMseqs2	92.19	85.66	62.76	41.36	96.94	92.90	74.76
MMseqs2 tax.*	94.33	86.56	67.47	43.44	98.11	93.47	75.09
minimap2	86.12	76.06	44.12	20.03	93.46	86.71	66.68
Kraken2	86.26	75.99	44.36	19.50	93.65	87.13	70.56
sourmash	30.69	9.04	25.14	3.48	31.04	8.00	3.07
DeepMicrobes	97.18	86.62	67.25	36.61	98.13	92.11	66.43
BERTax	95.65	83.88	90.06	54.10	98.62	95.10	66.92
MMseqs2+							
DeepMicrobes	99.33	93.67	76.91	50.51	97.23	93.20	77.85
BERTax	99.33	94.84	87.57	60.34	99.76	97.83	79.33
random	25.00	3.33	25.00	3.33	25.00	2.27	0.64

For a description of the datasets, see, for example, Fig. 2 above. Taxonomic classification ranks considered are superkingdom (Supk.), phylum (Phyl.), or genus (only final selection dataset). The best-performing tool is shown in dark gray, and the second best is shown in light gray. All corresponding confusion matrices are available at https://github.com/f-kretschmer/bertax.

For the closely related dataset, the AveP ranges for prediction of superkingdoms and phylum are from 30.69% and 9.04% (sourmash) to 97.18% and 86.62% (DeepMicrobes), respectively.

Additionally, we devised a combination of classic database approaches with machine learning methods, which increases the overall recall while preserving the precision, resulting in more useful taxonomic information. We first predict the taxonomy by querying the reference database using MMseqs2, as it is a method with a high proportion of predicted samples and high accuracy (*SI Appendix*, Table S2). We then predict the taxonomy of all unsuccessfully queried samples using BERTax or DeepMicrobes. This strategy limits the number of incorrectly predicted samples. Using this strategy, MMseqs2 + BERTax (99.33% and 94.84%) performs better than MMseqs2 + DeepMicrobes (99.33% and 93.67%). This is to be expected, as both MMseqs2 and DeepMicrobes depend on k-mers for their prediction. Therefore, samples that are difficult to classify for MMseqs2 tend to also be challenging to classify for DeepMicrobes, resulting in a limited higher precision. The combination of BERTax and MMseqs2 outperforms all competing tools.

Real-world use cases always comprise a mix of sequences with and without similar representatives in the reference database. Therefore, it is reasonable to use a combined approach of a database approach and BERTax, as the classification performance is likely better, both recall- and precision-wise.

When we compare the classification performances of different tools, we typically observe a trade-off between accuracy and the proportion of samples classified (*SI Appendix*, Table S2). If the classification tool uses a restrictive threshold, we observe a very precise classification. However, a lower number of sequences could be classified, whereas, when a high number of sequences is classified, usually, it comes with the drawback of a less precise classification. The proportion of classified sequences varies between 6.90% for sourmash and 92.69% for MMseqs2 taxonomy*. Neural network approaches like BERTax and DeepMicrobes always reach 100% due to their mode of operation without confidence thresholds.

For further analysis, see *SI Appendix*, Figs. S3 and S4, which show micro and macro averaged Precision–Recall (PR) curves to visualize the trade-off between accuracy and proportion of classified sequences (see also *SI Appendix*, Table S2). PR curves reflect the quality of prediction independently of the sensitivity of the tested tool. Additionally, for micro and macro average receiver operating characteristics (ROC) curves, see *SI Appendix*, Figs. S7 and S8.

BERTax Superior on De Novo Sequences. Unknown sequences are simulated by removing all sequences of one or several complete genera from the training set, and corresponding sequences are only used in the test dataset resulting in the distantly related dataset. For this dataset, BERTax outperforms, at ∼90% any other tool, ranging from 24.60 to 67.47% for superkingdom classification; see Table 1. Interestingly, MMseqs2 + BERTax performs not as well as BERTax alone, hinting at a not stringent enough classification of MMseqs2. Instead of not classifying a sequence and passing it to the DNN, a misclassification leads to worse results, which could be changed by using more-stringent standard parameters for MMseqs2. On the phylum level, BERTax (with or without MMseqs2) outperforms all other tools, albeit with a smaller margin than for superkingdoms. For both taxonomic ranks, DeepMicrobes performs worse than the best database approach, MMseqs2 taxonomy*.

We observe a drastic decrease in the proportion of predicted sequences for all classic homology-based approaches compared to the closely related dataset, with 52.78% for MMseqs2, 71.37% for MMseqs2 taxonomy*, 19.88% for minimap2, 21.56% for Kraken2, and 0.16% for sourmash (*SI Appendix*, Table S2). This was expected, since there are no sequences of the same genus in the target database, and thus fewer similar sequences are found.

Furthermore, when investigating sequence similarity in addition to taxonomic origin, removing all sequences from the test partition of the distantly related dataset which could be mapped to sequences in the training partition via BLAST (7), this observation is reinforced. While this filtering has only a minor effect on the rank superkingdom classification accuracy of BERTax (−0.65%), the accuracy of DeepMicrobes and MMseqs2 taxonomy*, drops by 5.15% and 5.99%, respectively (*SI Appendix*, Table S5). In creating the evaluation datasets, sequence similarity between the training and test partitions was not considered, in order to focus on the taxonomic aspects. However, when investigating the special case of predicting sequences with high sequence similarity in both partitions,

BERTax showed less dependence on sequence similarity than all other evaluated methods.

The results for this dataset are especially remarkable, since only a tiny fraction of organisms is so far described in databases, and the power of BERTax outperforming any other classification tool for de novo sequences appears to be of utmost importance to the metagenome community.

Performance of Final BERTax Model. We evaluated the final BERTax model in comparison to other methods on the final model dataset. This dataset differs from the closely related dataset in that the number of eukaryotic and bacterial fragments is almost tripled. This results in an increased number of closely related sequences in the reference database. An important difference from the models evaluated on the closely and distantly related dataset is that, additionally to superkingdom and phylum, an extra output layer is added to also predict the taxonomic genus of sequences. Expectedly, we observed higher AveP values for the final selection dataset compared to the closely related dataset (Table 1 and *SI Appendix*, Figs. S5, S6, S9, and S10). Comparing BERTax to the database approaches, it performs better for superkingdom (98.62%) and phylum (95.20%) prediction. For the prediction of the lowest taxonomic rank (genus), BERTax achieves a 66.92% comparable performance to DeepMicrobes and minimap2. Both deep learning approaches achieve significantly better results (+7% points) when comparing the class weight–normalized micro PR curve (*SI Appendix*, Fig. S5). This increase is caused by changes in the sample order in the micro averaged plot, indicating that both methods are impressively good at estimating their prediction performance. Interestingly, BERTax reached a higher AveP than DeepMicrobes for all taxonomic ranks. This is in contrast to the comparable, yet smaller, closely related dataset, indicating an architectural advantage when predicting more taxonomic ranks, while providing more training data.

The decrease in predictive ability for lower taxonomic ranks, also seen for the closely related dataset and distantly related dataset, might be due to the lower number of available training examples per class and the chosen architecture. Further, it is plausible that lower taxonomic ranks provide more advantages for the correct prediction of higher taxonomic ranks than vice versa (*SI Appendix*, Fig. S2).

The combined approach of MMseqs2 + BERTax is again the best performing one (Table 1 and *SI Appendix*, Figs. S5, S6, S9 and S10). On the final dataset, MMseqs2 + BERTax reached an increase in AveP of 0.43 percentage points for the rank superkingdom and almost three percentage points for the rank phylum (Table 1) compared to the closely related dataset.

Although only the numbers of eukaryotic and bacterial sequences were increased in the training data, for all four superkingdoms, the AveP of BERTax improved significantly, between 2.58 and 3.24 percentage points; see *SI Appendix*, Table S3. This broad increase in performance may be due to a greater amount of generally usable information. This information consists of more examples of each 3-mer from which its "general meaning" can be inferred, but also a larger number of examples of the context of each 3-mer. As a result, the 3-mer embeddings benefit, and the prediction performance increases.

It has to be emphasized that, although the final dataset is an extension of the closely related dataset—not excluding genera also present in the test set—we expect the generalization ability of BERTax to be comparable to that shown for the distantly related dataset with these restrictions, as the only substantial difference is the larger set of training data. However, even though BERTax clearly outperforms all other methods on the distantly related dataset, BERTax also naturally benefits greatly from more training data, especially taxonomically related sequences, as is evident from the performance comparison between datasets. This can be illustrated further when comparing the performance for specific taxonomic classes from the test set (see confusion matrices in *SI Appendix*, Fig. S13). Regarding the rank phylum, there are some classes for the distantly related set like *Crenarchaeota* or *Euryarchaeota* with near-perfect classification, but also classes like *Bacillariophyta* or *Deinococcus-Thermus* with a very low rate of correct classifications. These results are in contrast to those obtained from the final dataset, where performance differences between taxonomic classes are much lower. This shows that, for real-world use cases, it is most desirable for the training set to be taxonomically as complete as possible. However, this means that performance differences for specific taxonomic phyla or genera, as shown in *SI Appendix*, Fig. S13, are expected to be different for real-world applications; therefore, these differences have to be interpreted with extreme care. In conclusion, while we expect the final BERTax model to be as generalizable as shown with the distantly related dataset, the benefit of potentially having trained on sequences from similar taxonomic origin is high.

Contribution of Attention Heads to Performance. In order to quantify the importance of the attention mechanism for obtaining good predictions, we ran ablation studies varying both the number of attention heads per transformer block and the number of transformer blocks in the BERT part of BERTax (Fig. 3). These ablation studies were performed on the largest, final dataset but also on the distantly related dataset, to investigate generalizability. Results (*SI Appendix*, Table S7) show that reducing the number of attention heads per layer substantially decreases performance. On the other hand, reducing the number of transformer blocks shows ambivalent effects. Accuracy for the ranks phylum and genus (the latter is only predicted in the final dataset) increase, while superkingdom accuracy decreases for the distantly related dataset and shows little difference for the final dataset. All differences for this variation are, however, only minor. A possible explanation for this behavior could be that having more transformer layers leads to slightly better generalizability for the highest taxonomic rank, as, here, more abstraction capability is required. This, however, comes at the cost of slightly decreased accuracy for the prediction of lower taxonomic ranks.

Peeking into the Black Box. Although (deep) neural networks are a powerful tool, their behavior is often hard or impossible to interpret, resembling a "black box." However, as BERT, at its core, uses a (self-)attention mechanism, it is possible for us to gain insights into the feature relations and classification for given examples; see *SI Appendix*, Fig. S11, drawn with bertviz (https://github.com/jessevig/bertviz) (24). Analyzing all inferred syntactic and semantic relations of DNA tokens presupposed by the model would be beyond the scope of this paper. However, some general remarks can be made both for the relationship of different attention heads and for individual attention heads. In general, weight patterns show a high degree of heterogeneity between attention heads, suggesting that, similar to natural languages, each attention head represents a different kind of syntactic relationship between the tokens (25). Generic attention head patterns, as described in refs. 25 and 26, can be observed in BERTax, too: Head 4 in layer 1, for example, always attends to the previous token (*SI Appendix*, Fig. S11*A*), and head 4 in layer 2—albeit less clearly—attends to the next token (*SI Appendix*, Fig. S11*B*). In contrast, since (general genomic) DNA sequences do not have

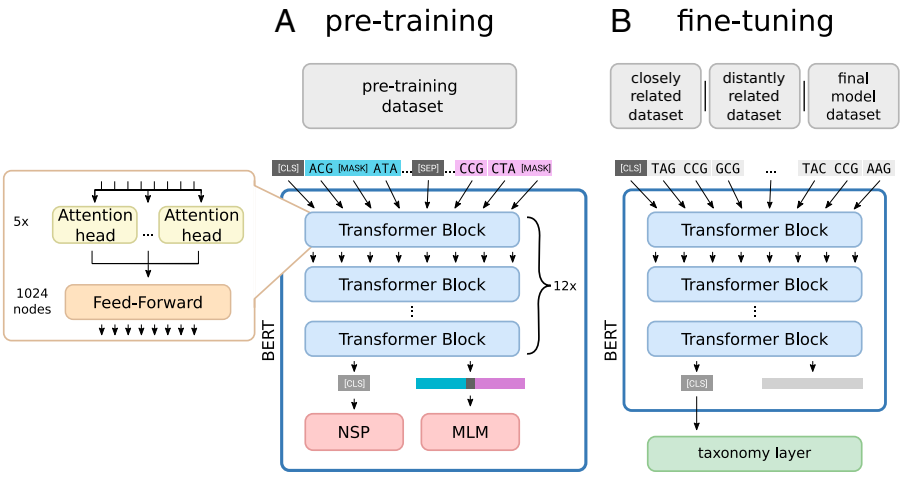

Fig. 3. The training process of the BERTax model. (*A*) During pretraining, two separate sentences are used as input. Twelve stacked transformer blocks process this input, each consisting of five attention heads that determine which parts of the inputs are relevant to each other, and a feed-forward layer that combines the results. The input, consisting of 502 tokens with 250 embedding dimensions each, is passed through all transformer blocks. Last, the NSP layer uses the classifier token [CLS] to predict whether the second sentence is an original successor sentence or a replaced sentence. The MLM layer predicts all masked tokens from the remaining tokens. The evaluation of MLM and NSP evaluates the pretraining of the transformer blocks, and thus the quality of the learned language. After pretraining, the pretrained transformer blocks are used for the fine-tuning step. (*B*) A single sentence is used as input during fine-tuning with one of the three datasets (closely related dataset—training and test data are phylogenetically closely related sequences; distantly related dataset—unknown genera are used as test dataset; and final model dataset—an expanded closely related dataset). The pretrained transformation blocks process this input, and only the [CLS] token is delivered to the taxonomy layers as output (*SI Appendix*, Fig. S1).

specific "separation" tokens comparable to periods in natural text, this kind of relationship is not displayed as an attention head pattern in BERTax. It can be noticed, however, that a very small number of tokens always aggregates the attention of all other tokens in specific attention heads (*SI Appendix*, Fig. S11 *D–F*). While these tokens are not the same between sequences, it can be argued that these tokens alongside the corresponding attention heads serve to define segments in the sequences, similar to how punctuation tokens are used in natural language models.

The biological interpretation of the working principle is very challenging, and clear conclusions remain difficult. Firstly, relationships similar in expressiveness to subject–object or subject–predicate in natural languages are hard to define. More importantly, however, in contrast to BERT models for natural languages, where tokens correspond to words (or word parts), no such counterpart exists in DNA sequences. The 3-mers, as used here, or any kind of k-mers, are always arbitrary units, especially when general genomic DNA sequences are considered. Furthermore, with only 64 different 3-mers, context becomes far more important as compared to natural languages with thousands of tokens. However, using the attention weights measure introduced in ref. 27 does lead to some interesting biological insights. For instance, examining BERTax attention weight patterns suggests that the model can distinguish between regions of strong and less strong selection pressure; see *SI Appendix*, Fig. S15. In particular, ribosomal RNA, if included in the query sequence, appears to be an important region for classification (*SI Appendix*, Fig. S14). The importance of the different 3-mer types is very different. This suggests that there are some 3-mers whose distribution is more likely to allow inferences about taxonomy than others. For further details on attention weights, see *SI Appendix, Attention weight statistics*. We also found that the importance of homologous sequences varies considerably across superkingdoms. Homologous sequences are likely to be particularly advantageous at a low taxonomic level. Thus, the advantage of a homologous sequence could be related to the number of phyla and genera in the superkingdom being determined; see *SI Appendix*, Table S6.

To really understand the decision-making process in BERTax, a more thorough analysis beyond the scope of this paper is necessary.

Discussion

Even today's most comprehensive sequence databases miss significant portions of the total biodiversity. This incompleteness can result in large proportions of taxonomically unclassifiable DNA sequences. This is a common problem when using classical approaches, based on similarity to known sequences, on environmental samples. Here we present BERTax to tackle this task. BERTax is a tool for taxonomic classification of DNA sequences (reads, contigs, or scaffolds) using DNNs. The ranks considered are superkingdom, phylum, and genus. The focus and strength, however, of BERTax lies in a more general taxonomic classification, which is reflected by the difference in performance gain for different taxonomic ranks; the most substantial improvement is made for the rank superkingdom.

To the best of our knowledge, BERTax is the only taxonomic classification method not based on local similarities between the query and the target or limited to the use of k-mer frequencies, as even other deep learning methods such as DeepMicrobes are. This provides a likely explanation for the superior generalization ability of BERTax as compared to other tools.

In general, when developing machine learning methods, avoiding the problem of data leakage is crucial. Data leakage happens when a model learns shared information between training and testing data not available during actual use—in other words, the model "cheats." We have put particular emphasis on making sure this is not the case with BERTax. Most importantly, dataset splits into training and testing data were not allowed to include duplicate samples; sequences were filtered using a sequence similarity threshold to avoid highly similar sequences shared between the splits. Incidentally, this also means that whole reference genomes cannot be present in the training data, which presents a necessary deviation from the designed use case of database methods compared against. Common pitfalls, such as providing metadata for each sample, were omitted by exclusively using the DNA sequence as input. Furthermore, allowing the model to

learn classification based on sequencing quality was avoided by imposing corresponding thresholds when creating the datasets. On the other hand, shared homologous segments between sequences of the training and testing splits do not constitute data leakage, but present valid taxonomic features, used as the foundation of all database classification tools compared against in the evaluation. Nevertheless, even the influence of (potential) shared homology on the performance of BERTax is minor, as was shown.

While our method has a higher AveP than comparable methods, a combination of a sensitive classic approach, like MMseqs2, and BERTax further improves the prediction performances, reaching unmatched sensitivity and specificity. The great power of BERTax can be observed especially for sequences with no closely related species in the reference database or training dataset.

Due to its high sensitivity and specificity, relatively independent of closely related species in the training data, BERTax is very suitable for use with metagenomic samples, especially on sequences that could not be classified with database methods. Thus, BERTax reduces the "microbial dark matter" and promises a significant benefit for metagenomics. In general, the applications of BERTax are manifold, as it can be applied in diagnostics to bypass otherwise lengthy cultivation processes, for example, in the diagnostics of fungi. Here, sequencing can be used together with BERTax to classify very rapidly directly at read level to identify potential infections. For the detection of endogenous viral elements (EVEs), the windowed mode of BERTax can be used to analyze target genomes. All windows classified as viruses can be considered as potential EVEs.

Alongside other publications (28–31), our results yet again emphasize the power of natural language processing methods in the field of biological sequences. Furthermore, a self-attention-based method like BERT grants a different perspective on the structure of DNA sequences, allowing—in theory—investigation of how the network comes to its classification "decision." However, this process is very labor intensive in practice and rarely results in major discoveries.

Materials and Methods

BERTax is based on the architecture BERT, used in many natural language processing tasks (21). BERT appears to outperform previous NLP methods, due to the division of the training process into unsupervised pretraining and supervised fine-tuning, and due to its deeply bidirectional nature of language processing (20). Deeply bidirectional means that both reading directions are considered simultaneously, as opposed to the "shallowly" bidirectional approach where the model is trained on both directions independently, as used in the competing method ELMO (embeddings from language models) (29, 32).

Pretraining: Learning the "DNA Language". During pretraining, the model is trained on two tasks that are not specific to the actual classification objective, but are rather designed to enable the model to learn general structural features of the language (or DNA, in our case). These tasks are masked language modeling (MLM) and next sentence prediction (NSP). MLM masks individual tokens (in NLP, a token is a word or a part of a word) which are then to be predicted. NSP predicts whether the second of two sentences is related to the first or whether it was replaced by a random independent sentence. When training the BERT model, MLM and NSP are trained together, with the goal of minimizing the combined loss function of the two strategies (20).

To model DNA as natural language, BERTax was pretrained on DNA sequences of a fixed length of 1,500 nucleotides. Each sequence was split into 500 3-mers (i.e., tokens in a natural language), with the first 250 3-mers being the first sentence and the second 250 3-mers being the second sentence (*SI Appendix*, Fig. S12). For each input, 15% of the 3-mers were masked for MLM to learn what token fits which context, and, for 50% of the inputs, the second sentence was randomly replaced by a sentence from another DNA fragment as a negative training set for NSP only.

Internally, a 3-mer is a token (i.e., a string with an assigned meaning), with 64 possible tokens assuming the occurrence of the four canonical nucleotides. BERT uses, additionally, five specific tokens necessary for training: the unknown token [UNK], representing all 3-mers containing at least one ambiguously sequenced character, such as "N"; the empty token [PAD] which is required for padding shorter input sequences to the required input length of 500 tokens; the classification token [CLS], which is designed to represent the "meaning" of the entire sentence; the mask token [MASK], which masks the words to be predicted in MLM; and the separator token [SEP], which separates the two sentences for the NSP task.

On the highest hierarchical level, the BERT architecture consists of a specified number of layers called transformer blocks (Fig. 3*A*). Each transformer block contains a certain number of attention heads–where weights are learned using the mechanism of self-attention–and one feed-forward layer, which serves as the connection between the transformer blocks. Hyperparameters, that is, parameters whose values are used to control the learning process, include the number of transformer blocks (set to 12), the number of attention heads per block (5), and the size of the feed-forward layers (1,024). The embedding dimension, that is, the dimension of the internal representation of the sequences, is set to 250 (*SI Appendix*, Fig. S12). The dropout rate (fraction of nodes not trained per epoch) of the feed-forward layers, used for better generalization, is set to 5%. All hyperparameters are set to lower values than those used in the original BERT models (20). The main reason for this is that the vocabulary size for the 3-mer DNA sequences is much lower–65 as opposed to ∼30,000 for the English language tokens used in the original BERT models (33).

The length of the input is 502 tokens, composed of 500 tokens from the DNA sequence and two architecture-specific tokens to keep the sentences apart (*SI Appendix*, Fig. S12). This input is passed through all transformer blocks (Fig. 3). As typical for an encoder architecture like BERT, the dimensionality of the sequence passed between the blocks and layers stays the same. The output of the pretraining architecture comprises the NSP layer (i.e., the prediction whether or not the second sentence is a random replacement) and the MLM layer with one output per input position (502) and distinct token (69) for predicting the 3-mer of a masked position.

Data. Archaean and eukaryotic genomes were retrieved from NCBI using ncbi-genome-download (https://github.com/kblin/ncbi-genome-download/, version 0.2.12); viral and bacterial genomes were downloaded from NCBI manually. The list of genomes, including the respective assembly versions, is provided as Dataset S1.

Pretraining Dataset. For each superkingdom, we extracted 1 million fragments of length 1,500 nt from the downloaded genomes at uniformly distributed starting positions. To obtain a taxonomically balanced dataset, we grouped the genomes by the taxonomic rank "order" and extracted fragments evenly distributed from these groups.

More specifically, fragments were extracted in an iterative process. Examining each superkingdom individually, in a first step, species belonging to the respective superkingdom were grouped by their taxonomic order. Orders were then iterated upon, selecting one random species of the respective order in each iteration. For the selected species, the genomic position for extracting a fragment was also chosen randomly. Fragments with a high number of ambiguously sequenced characters (e.g., "N") were discarded and replaced. This threshold was set to more than one ambiguous base. The whole iterative process was repeated until 1 million fragments were reached.

To reduce redundancy, we clustered fragments by sequence identity using MMseqs2-linclust version 11.e1a1c (34) with a threshold of 80%, resulting in 939,357 eukaryotic, 764,161 bacterial, 535,153 archaean, and 253,803 viral fragments (Table 2, pretraining dataset). The varying degree of reduction across superkingdoms can be explained by 1) the different number of taxa in each superkingdom, 2) the number of genomes in databases, and 3) the genome sizes. We refer to this dataset as the pretraining dataset, used for pretraining of BERTax. As pretraining is unsupervised, the target classes of the taxonomic classification objective are not used, and model weights are trained solely on the generic tasks MLM and NSP.

Table 2. The development and the number of fragments/samples of the datasets

	Superkingdom			
	Eukaryotes	Bacteria	Archaea	Viruses
No. extracted fragments	1,000,000	1,000,000	1,000,000	1,000,000
After clustering	939,357	764,161	535,153	253,803
Resulting dataset: Pretraining				
After phylum selection	873,873	690,984	535,153	228,574
After partitioning*				
Closely related selection	873,873	690,984	535,153	228,574
Distantly related selection	854,524	684,813	532,214	227,265
Resulting dataset: Closely and distantly related				
No. extracted fragments	2,995,950	3,000,009	1,000,000	1,000,000
After clustering	2,707,781	1,903,183	535,153	253,803
After clade selection	2,707,781	1,903,183	535,153	253,803
After partitioning*	2,707,781	1,903,183	535,153	253,803
Resulting dataset: final model				

We extracted 1 million taxonomically balanced fragments of length 1,500 nt from the downloaded genomes for each superkingdom. The resulting fragments were clustered by 80% sequence identity, forming the pretraining dataset used as input for pretraining. Next, we selected, from the pretraining dataset, all phyla with at least 10,000 fragments. We partitioned these fragments into a test set (2,000 sequences per phylum), training set (95% of nontest set), and validation set (5% of nontest set). For this, we used two different approaches, closely related selection and distantly related selection (see Figs. 3 and 4), resulting in two different datasets, the closely related dataset and the distantly related dataset. Since the genomic diversity of eukaryotes (939,357) and bacteria (764,161) seems to be less covered compared to archaea (535,153) and viruses (253,803) (see pretraining dataset), we extracted another 2 million fragments and clustered again by sequence similarity. Next, we selected all superkingdoms, phyla, and genera with at least 10,000 fragments. However, all clades with less than 10,000 fragments were combined into a new class, "unknown," for each taxonomic rank. Therefore, the number of fragments did not decrease. All fragments were subsequently divided via closely related selection into test set, training set, and validation set (see proportions above). This process results in the final-model dataset.
*Into training, validation, and test sets.

Preparing Data for Fine-Tuning. For fine-tuning, we sorted the fragments from the pretraining dataset into classes according to the phylum the fragment originated from. We selected only classes with at least 10,000 fragments. All fragments of the selected classes form the input data, which we refer to as "samples," as is common in machine learning. This results in 30 phylum classes with 2,328,584 samples in total. We partition these data into training, validation, and test sets. The training set is used to train the neural network. The validation set is used to prevent overfitting, by evaluating the neural network's loss (the distance to the optimal solution) after each training epoch, which allows "early stopping," that is, terminating training when the loss of the validation set does not further decrease. The test set is used to evaluate the performance of the neural network after finishing the training.

Three Evaluation Datasets. We used three different evaluation datasets. The first and second datasets are compared to show the power of BERTax when unknown sequences are part of the test set. This is achieved by using different strategies to partition our data into training, validation, and test sets (Fig. 4).

When creating the evaluation datasets, we did not exclude homologous sequences between the training and the test datasets. This is necessary for a realistic comparison with methods based on sequence similarity. When removing homologous sequences, methods like local alignments have fewer hits and would be at a disadvantage. The presence of homologous sequences allows the trained BERTax model to remember sequence segments. These can then be used to classify similar sequences with the same taxonomy. We do not consider this behavior to be critical, since methods based on sequence similarity already established this functionality. The extent of influence of homologous on the prediction quality can be found in more detail in *SI Appendix*, Table S5.

The first evaluation dataset is called the closely related dataset. As a test set, we randomly select 2,000 samples per phylum. From the remaining samples, we select 95% of the samples of each phylum as the training set and use the remaining 5% as the validation set (Fig. 4*A*).

The second evaluation dataset is called the distantly related dataset. The goal of distantly related selection is to simulate sequences without (taxonomically) closely related sequences in the reference database, simulating "unknown" sequences. For this purpose, a genus separation between training and test sets is employed; that is, for no sample in the test set does there exist a sample of the same genus in the training set (Fig. 4*B*). To achieve this, we split each phylum into its genera. As a test set, we select a subset of these genera with about 2,000 samples–more precisely, the subset with the number of samples closest to 2,000. The test set is balanced by undersampling randomly: We keep 1,780 samples per phylum (according to the smallest subset). From the remaining samples in each phylum, we again select 95% for the training set and use the remaining 5% as the validation set.

The third evaluation dataset is referred to as the final model dataset. The initial redundancy of the extracted eukaryotic and bacterial fragments is very low, indicated by the low reduction of fragments after clustering the pretraining dataset (Table 2). Therefore, the genomic diversity of the two superkingdoms is probably underrepresented. Thus, for fine-tuning our final model which is used in the downloadable version of BERTax, we extracted an additional 2 million fragments for those two superkingdoms, clustered again by sequence similarity. With this dataset, a more complete snapshot of the genomes is provided, with a wider textual and taxonomic diversity.

For this dataset, we sorted the fragments into classes according to the superkingdom, phylum, and genus the fragment originated from. Again, we selected only classes with at least 10,000 fragments. Remaining classes are moved to an additional class "unknown," which is introduced for each taxonomic rank (i.e., "unknown superkingdom," "unknown phylum," "unknown genus"). With this, the number of classes per taxonomic rank is five for the rank superkingdom, 44 for phylum, and 156 for genera. We partition this dataset into training, validation, and test sets using closely related selection as described above (Fig. 4*A*).

Fine-Tuning: Taxonomic Classification. Fine-tuning is used for the training of the pretrained model on the problem of interest. In our case, this is the prediction of phylogenetic taxa, by classification of sequences into different classes, representing different taxa. While fine-tuning, each sample is converted from its full length of 1,500 nucleotides into the corresponding 500 tokens and the classification token [CLS] (representing the whole sequence). Because the required sequence length is 502, one additional padding token ([PAD]) has to be added to the end. The pretrained transformer blocks process these 502 tokens, and, from the resulting output, the [CLS] token is used by the taxonomy layer/s to predict the taxonomy (Fig. 3*B* and *SI Appendix*, Fig. S1). Next, the weights and biases of all layers, including the pretrained transformer blocks, are adapted for a more precise prediction given this specific input.

We fine-tuned all models for a maximum of 16 epochs, employing early stopping (see *SI Appendix*, Table S4 for the exact number of epochs trained). To avoid bias toward predicting the most frequent classes, we balanced the classes for fine-tuning by class weights, calculated for each taxonomic rank,

$$w_i = \frac{\sum_{j=1}^{C} n_j}{C \cdot n_i}.$$

Here, w_i is the class weight, and n_i is the number of samples for class $i = 1, \ldots, C$, applied to each sample. Using these weights allows preserving the

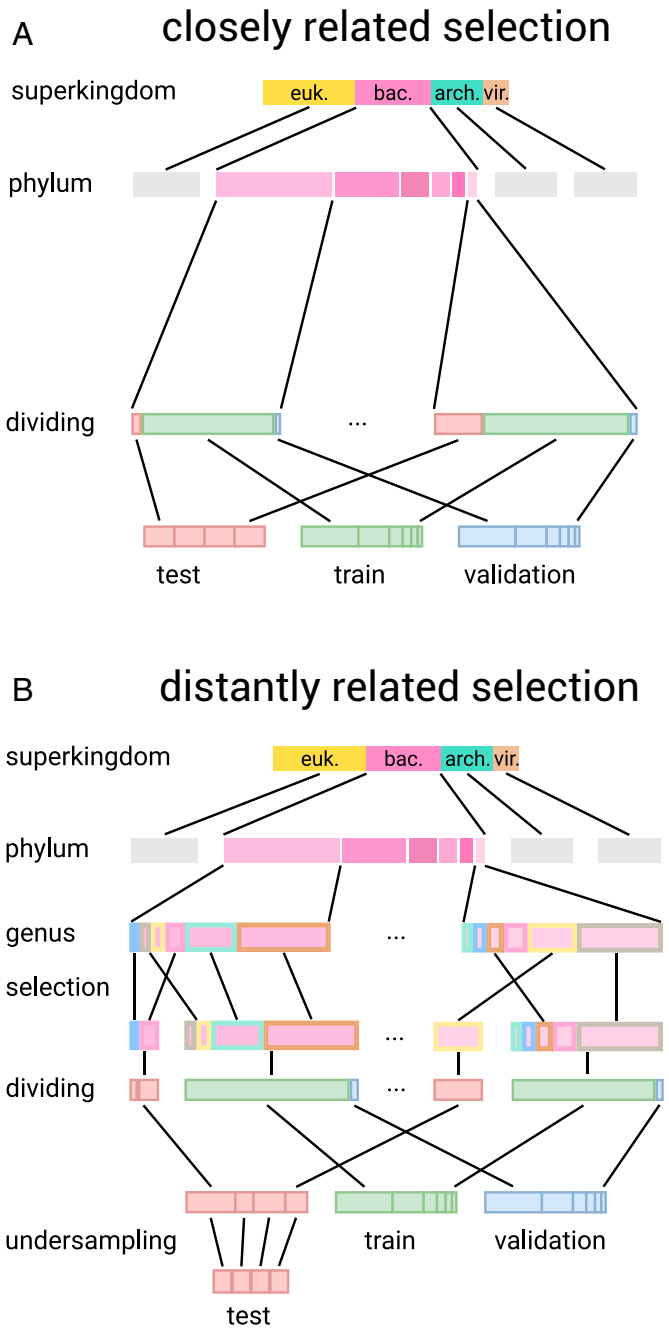

Fig. 4. Data preparation of the test, training, and validation set via closely related selection and distantly related selection. (*A*) The closely related selection uses 2,000 randomly selected samples per phylum as a test set. All remaining samples of each phylum are divided into 95% as training set, and the remaining 5% are used as the validation set; euk., eukaryota; bac., bacteria; arch., archaea; vir., viruses (*B*) The distantly related selection selects, per phylum, one or more entire genera as the test set (in comparison to the closely related selection, where no restriction in this regard is made). This subset is the combination of genera with the number of samples closest to 2,000. The test set is balanced by undersampling, reducing the size per phylum to the size of the smallest subset. As for the closely related selection, all remaining samples of each phylum are divided into 95% as the training set and the remaining 5% as the validation set.

complete number of samples while ensuring that the influence of each class on the fine-tuning is balanced.

Ideally, the representative proportions of each superkingdom would be used for training, without using class weights. This would bias the model classification toward the expected taxonomic distribution. However, to the best of our knowledge, it is not clear what these representative proportions are. Likely, these proportions are highly dependent on the origin of the sample and can therefore vary considerably. Therefore, we preferred to choose a neutral training and did not build in a bias from the beginning.

Architectures. We tested three different architectures based on the pretrained BERT model, which differ in the adaptations of the output layers (*SI Appendix*, Fig. S1). The pretrained model is used to obtain a learned high-dimensional embedding of the input sequence. Specifically, the classification token [CLS], which is a single vector designed to contain the information of the whole input sequence, is the input to the new output layer. Each output layer uses the activation function softmax, which results in a probability distribution for each taxonomic rank, such that each prediction is associated with a confidence.

The flat architecture (*SI Appendix*, Fig. S1*A*) is trained to directly predict the lowest taxonomic rank, which is the phylum in the comparison of the architectures. The flat architecture uses a simple dense layer with a node for each phylum. The nested architecture (*SI Appendix*, Fig. S1*B*) consists of multiple independent BERT models arranged in a tree-like manner. Each of these models is trained only on samples of its taxonomic group and uses a dense layer containing the same number of nodes as its subclasses. First, the root model predicts the superkingdom. Then, the sample is passed to the model of the subclass with the highest probability. This is continued until the lowest taxonomic rank is reached. The all-in-one architecture (*SI Appendix*, Fig. S1*C*) is a single BERT model which predicts all taxonomic ranks simultaneously. It uses the idea of Rojas-Carulla et al. (35) to provide the prediction of higher taxonomic ranks to lower prediction layers. For this, all output layers (taxonomic ranks) of the model have access to the BERT model itself and the output layers of all higher taxonomic ranks.

The unique characteristic of the all-in-one architecture is the additional information that output layers have access to: The superkingdom output layer is connected to the phylum layer, potentially allowing the prediction of the superkingdom to benefit from (high-level) features learned to be important for phylum classification (also see *SI Appendix*, Fig. S2).

Comparison to Other Tools. BERTax is compared against the state-of-the-art database taxonomic classification approaches Kraken2 (15), sourmash (16), MMseqs2 (14), and minimap2 (17). Hereby, Kraken2 and sourmash use *k*-mers and minimizers for comparing the query to the reference database. MMseqs2 uses *k*-mers and local alignments. Minimap2 uses minimizers for the identification of seeds, which are further extended in a local alignment. For all tools, we used the default parameters. However, we use a modified version of MMseqs2 taxonomy. More precisely, we are using MMseqs2 with the parameters of MMseqs2 taxonomy (–e-profile 0.001, -e 1). Doing so, we get an E value and thus a significance value that can be used to calculate the PR curve and AveP, rather than assigning the same confidence to all predictions. With this approach, we receive the same hits as for MMseqs2 taxonomy for 99.94% of all samples. We call this approach MMseqs2 taxonomy*. As a state-of-the-art machine learning approach, we tested DeepMicrobes (19). In our evaluation, we used the architecture and hyperparameters evaluated as best by Liang et al. (19). The architecture comprises long short-term memory layers with self-attention that use *k*-mer embeddings ($k = 12$) as input. The DeepMicrobes models are trained on exactly the same data as BERTax.

The deep learning–based method GeNet (35), developed for classifying bacteria, unfortunately could not be compared against, as it relies on downloading and subsequently binarizing training data on its own, which is highly impractical for our much bigger datasets. For Convolutional Neural Network - Relative Abundance Index (CNN-RAI) (36), only the source code of the tool is provided. Therefore, it is impossible to train the method on new data with reasonable effort, which is necessary for comparability.

Seq2Species (37) and MetagenomicDC (38) restrict the input to 16S sequences, which severely limits the approaches' applicability; these tools are therefore not included in our comparisons. The same applies to CHEER (39) which only features RNA virus taxonomy classification. However, most of these approaches use convolutional neural networks, which use combinations of short letter sequences (3 nt to 12 nt) for classification. This is similar to the use of *k*-mers in database approaches.

Downloadable BERTax Version. The downloadable BERTax version is built on the all-in-one architecture and trained on the final model dataset. The downloadable tool additionally predicts the taxonomic rank genus.

BERTax was implemented in Python 3.7 and uses the Python packages scipy (1.6.1) (40), keras (2.4.3), tensorflow (2.4.1) (41), numpy (1.19.2) (42), and keras-bert (0.86.0). The visualization feature is based on bertviz (1.0.0). The

source code as well as a conda package and docker container are available at https://github.com/f-kretschmer/bertax.

Data, Materials, and Software Availability. Source code of the final BERTax tool and data used in training the neural networks have been deposited in https://github.com/f-kretschmer/bertax (43) and the Open Science Framework (OSF) (10.17605/OSF.IO/QG6MV) (44).

ACKNOWLEDGMENTS. This work was funded by the Ministry for Economics, Sciences and Digital Society of Thuringia, under the framework thurAI (Grant 2021 FGI 0009), project "Mikrobiom-Analyse"; under the framework of the Landesprogramm ProDigital (Grant DigLeben-5575/10-9); and Collaborative Research Centre 1076 AquaDiva. The funders had no role in study design, data collection and analysis, decision to publish, or preparation of the manuscript.

1. C. Mora, D. P. Tittensor, S. Adl, A. G. Simpson, B. Worm, How many species are there on Earth and in the ocean? *PLoS Biol.* **9**, e1001127 (2011).
2. B. B. Larsen, E. C. Miller, M. K. Rhodes, J. J. Wiens, Inordinate fondness multiplied and redistributed: The number of species on earth and the new pie of life. *Q. Rev. Biol.* **92**, 229–265 (2017).
3. S. Louca, F. Mazel, M. Doebeli, L. W. Parfrey, A census-based estimate of Earth's bacterial and archaeal diversity. *PLoS Biol.* **17**, e3000106 (2019).
4. J. J. Wiens, Vast (but avoidable) underestimation of global biodiversity. *PLoS Biol.* **19**, e3001192 (2021).
5. S. Louca, F. Mazel, M. Doebeli, L. W. Parfrey, Response to "vast (but avoidable) underestimation of global biodiversity". *PLoS Biol.* **19**, e3001362 (2021).
6. C. Brandt, E. Bongcam-Rudloff, B. Müller, Abundance tracking by long-read nanopore sequencing of complex microbial communities in samples from 20 different biogas/wastewater plants. *Appl. Sci. (Basel)* **10**, 7518 (2020).
7. A. Morgulis *et al.*, Database indexing for production MegaBLAST searches. *Bioinformatics* **24**, 1757–1764 (2008).
8. B. Buchfink, C. Xie, D. H. Huson, Fast and sensitive protein alignment using DIAMOND. *Nat. Methods* **12**, 59–60 (2015).
9. R. Ounit, S. Wanamaker, T. J. Close, S. Lonardi, CLARK: Fast and accurate classification of metagenomic and genomic sequences using discriminative k-mers. *BMC Genomics* **16**, 236 (2015).
10. V. K. Bui, C. Wei, CDKAM: A taxonomic classification tool using discriminative k-mers and approximate matching strategies. *BMC Bioinformatics* **21**, 468 (2020).
11. D. Kim, L. Song, F. P. Breitwieser, S. L. Salzberg, Centrifuge: Rapid and sensitive classification of metagenomic sequences. *Genome Res.* **26**, 1721–1729 (2016).
12. P. Menzel, K. L. Ng, A. Krogh, Fast and sensitive taxonomic classification for metagenomics with Kaiju. *Nat. Commun.* **7**, 11257 (2016).
13. D. Ainsworth, M. J. E. Sternberg, C. Raczy, S. A. Butcher, k-SLAM: Accurate and ultra-fast taxonomic classification and gene identification for large metagenomic data sets. *Nucleic Acids Res.* **45**, 1649–1656 (2017).
14. M. Steinegger, J. Söding, MMseqs2 enables sensitive protein sequence searching for the analysis of massive data sets. *Nat. Biotechnol.* **35**, 1026–1028 (2017).
15. D. E. Wood, J. Lu, B. Langmead, Improved metagenomic analysis with Kraken 2. *Genome Biol.* **20**, 257 (2019).
16. C. T. Brown, L. Irber, sourmash: A library for MinHash sketching of DNA. *J. Open Source Softw.* **1**, 27 (2016).
17. H. Li, Minimap2: Pairwise alignment for nucleotide sequences. *Bioinformatics* **34**, 3094–3100 (2018).
18. S. H. Ye, K. J. Siddle, D. J. Park, P. C. Sabeti, Benchmarking metagenomics tools for taxonomic classification. *Cell* **178**, 779–794 (2019).
19. Q. Liang, P. W. Bible, Y. Liu, B. Zou, L. Wei, Deepmicrobes: Taxonomic classification for metagenomics with deep learning. *NAR Genom. Bioinform.* **2**, lqaa009 (2020).
20. J. Devlin, M. W. Chang, K. Lee, K. Toutanova, BERT : Pre-training of deep bidirectional transformers for language understanding. arXiv [Preprint] (2019). https://doi.org/10.48550/arXiv:1810.04805 (Accessed 10 June 2022).
21. I. Tenney, D. Das, E. Pavlick, Bert rediscovers the classical NLP pipeline. arXiv [Preprint] (2019). https://doi.org/10.48550/arXiv:1905.05950 (Accessed 14 March 2022).
22. A. Vaswani *et al.*, Attention is all you need. arXiv [Preprint] (2017). https://doi.org/10.48550/arXiv:1706.03762 (Accessed 11 May 2022).
23. N. A. O'Leary *et al.*, Reference sequence (RefSeq) database at NCBI: Current status, taxonomic expansion, and functional annotation. *Nucleic Acids Res.* **44**, D733–D745 (2016).
24. J. Vig, Visualizing attention in transformer-based language representation models. arXiv [Preprint] (2019). https://doi.org/10.48550/arXiv:1904.02679 (Accessed 20 April 2022).
25. K. Clark, U. Khandelwal, O. Levy, C. D. Manning, What does BERT look at? An analysis of BERT's attention. arXiv [Preprint] (2019). https://doi.org/10.48550/arXiv:1906.04341 (Accessed 20 April 2022).
26. O. Kovaleva, A. Romanov, A. Rogers, A. Rumshisky, Revealing the dark secrets of BERT. arXiv [Preprint] (2019). https://doi.org/10.48550/arXiv:1908.08593 (Accessed 21 April 2022).
27. H. Chefer, S. Gur, L. Wolf, "Transformer interpretability beyond attention visualization" in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition* (Institute of Electrical and Electronics Engineers, 2021), pp. 782–791.
28. Q. Zou, P. Xing, L. Wei, B. Liu, Gene2vec: Gene subsequence embedding for prediction of mammalian N^6-methyladenosine sites from mRNA. *RNA* **25**, 205–218 (2019).
29. M. Heinzinger *et al.*, Modeling aspects of the language of life through transfer-learning protein sequences. *BMC Bioinformatics* **20**, 723 (2019).
30. Y. Ji, Z. Zhou, H. Liu, R. V. Davuluri, Dnabert: Pre-trained bidirectional encoder representations from transformers model for DNA-language in genome. bioRxiv [Preprint] (2020). https://doi.org/10.1101/2020.09.17.301879 (Accessed 10 October 2021).
31. T. Bepler, B. Berger, Learning the protein language: Evolution, structure, and function. *Cell Syst.* **12**, 654–669.e3 (2021).
32. M. Zaib, Q. Z. Sheng, W. Emma Zhang, "A short survey of pre-trained language models for conversational AI-A new age in NLP" in *Proceedings of the Australasian Computer Science Week Multiconference, ACSW '20* (Association for Computing Machinery, New York, NY, 2020).
33. S. Zhao, R. Gupta, Y. Song, D. Zhou, "Extremely small BERT models from mixed-vocabulary training" in *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, P. Merlo, J. Tiedemann, R. Tsarfaty, Eds. (Association for Computational Linguistics, 2021), pp. 2753–2759.
34. M. Steinegger, J. Söding, Clustering huge protein sequence sets in linear time. *Nat. Commun.* **9**, 2542 (2018).
35. M. Rojas-Carulla *et al.*, Genet: Deep representations for metagenomics. arXiv [Preprint] (2019). https://doi.org/10.48550/arXiv:1901.11015 (Accessed 8 November 2021).
36. M. A. Karagöz, O. U. Nalbantoglu, Taxonomic classification of metagenomic sequences from relative abundance index profiles using deep learning. *Biomed. Signal Process. Control* **67**, 102539 (2021).
37. A. Busia *et al.*, A deep learning approach to pattern recognition for short DNA sequences. BioRxiv [Preprint] (2019) https://doi.org/10.1101/353474.
38. A. Fiannaca *et al.*, Deep learning models for bacteria taxonomic classification of metagenomic data. *BMC Bioinformatics* **19**, 198 (2018).
39. J. Shang, Y. Sun, CHEER: HierarCHical taxonomic classification for viral mEtagEnomic data via deep leaRning. *Methods* **189**, 95–103 (2021).
40. P. Virtanen *et al.*; SciPy 1.0 Contributors, SciPy 1.0: Fundamental algorithms for scientific computing in Python. *Nat. Methods* **17**, 261–272 (2020).
41. M. Abadi *et al.*, TensorFlow : Large-scale machine learning on heterogeneous systems, version: 2.6.0. https://www.tensorflow.org. Accessed 15 April 2022.
42. C. R. Harris *et al.*, Array programming with NumPy. *Nature* **585**, 357–362 (2020).
43. F. Kretschmer, F. Mock, BERTax: Taxonomic Classification of DNA sequences. GitHub. https://github.com/f-kretschmer/bertax. Deposited 12 June 2022.
44. F. Mock, F. Kretschmer, Supplement Paper: BERTax: taxonomic classification of DNA sequences with Deep Neural Networks. OSF. https://osf.io/QG6MV/. Deposited 7 October 2021.